Midterm Exam
CS 122A
Spring 2013
Max. Points: 100

(Please read the instructions carefully)

Instructions:
- The total time for the exam is 75 minutes; be sure to budget your time accordingly.
- The exam is closed book and closed notes.
- Be sure to answer each part of each question after reading them carefully.
- If you don’t understand something, ask the instructor for clarification.
- If you still find ambiguities in a question, note the interpretation you are taking.
- The last page of this exam is blank; you can use it as scratch paper.

![Comic Strip](image.png)

NAME: I. M. WRIGHT  STUDENT ID: 1

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Question 1: E-R Modeling (20 points)

(a) (20 pts) You have been hired as a part-time consultant by a new Orange County startup, VideoTweets, Inc., to help with the data management aspects of a new online service. The VideoTweets business plan is to do for videos what Instagram did for photos and then to be acquired by Facebook for $1B and to retire in the Caymans by the end of 2015. They plan to store and serve videos to mobile users from a cluster of Linux servers, with the videos themselves being stored in Linux files. They want to use a relational DBMS (DB2) to keep track of their users, the metadata about the videos, etc. Design and draw an E-R schema for their collection of data. Be sure to capture all relevant entities, relationships, and attributes. Clearly label each relationship to indicate what kind it is (e.g., 1:1, 1:N, etc.), what its participation constraints are, and so on. Clearly identify the primary key for each entity set (using a * or an underline). If you make any additional assumptions, note them next to your diagram. Here is what you have been told about the data by their user-facing application developers:

- Each user has a unique user name, an e-mail address, a name, and an age.
- Users can become video sharing friends by “liking” one another.
- Users can post videos and the posting date should be remembered. Any given video will have been posted by one user, and each video must in fact have some associated user who posted it.
- Videos have a unique id, a caption, the date when it was taken, a thumbnail preview, and a file name (simply a string containing the Linux filesystem path for the actual bits of the video).
- A user can be tagged in one or more videos, and a video can have multiple tagged users.
- To avoid use of the service by social outcasts, every user must be liked by at least one other user.
Question 2: E-R Translation and Semantics (30 points)

(a) (20 pts) Design a relational schema to represent the data described in E-R form below. If there are opportunities to avoid excess relations stemming from relationships, be sure to avoid them in order to make the relations simpler for querying. Clearly list all of the relations, attributes, primary keys, foreign keys (and their referenced target relations), and not null constraints so that your design represents the E-R model as faithfully as possible.

![E-R Diagram]

<table>
<thead>
<tr>
<th>Tables and their columns T(c1, c2, c3, ...)</th>
<th>Primary key column(s)</th>
<th>Foreign key(s) and their target tables</th>
<th>Not null columns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PERSON</strong> (ssno, name)</td>
<td>ssno</td>
<td>—</td>
<td>ssno</td>
</tr>
<tr>
<td><strong>PATIENT</strong> (ssno, addr, age, docssno)</td>
<td>ssno</td>
<td>ssno, docssno</td>
<td></td>
</tr>
<tr>
<td><strong>DOCTOR</strong> (ssno, medschool, specialty)</td>
<td>ssno</td>
<td>ssno, ssno</td>
<td></td>
</tr>
<tr>
<td><strong>DRUG</strong> (dname, formula)</td>
<td>dname</td>
<td>—</td>
<td>dname</td>
</tr>
<tr>
<td><strong>PHARMACY</strong> (pname, addr, phone)</td>
<td>pname</td>
<td>dname</td>
<td>pname</td>
</tr>
<tr>
<td><strong>SELLS</strong> (dname, pname, price)</td>
<td>(dname, pname)</td>
<td>dname, pname</td>
<td></td>
</tr>
<tr>
<td><strong>PRESCRIPTION</strong> (pssno, dname, docssno)</td>
<td>(pssno, dname)</td>
<td>pssno, docssno</td>
<td></td>
</tr>
</tbody>
</table>

SCORE: 20
Question 2: E-R Translation and Semantics (cont.)

(b) (10 pts) The following is a list of statements related to the E-R model above. For each of these statements indicate whether or not it is implied by the E-R model. Circle “Yes” if the E-R model indeed implies the given statement and circle “No” if it does not.

Yes  No  The price of a particular drug depends on which pharmacy is selling it.
Yes  No  A patient is uniquely identified by their social security number.
Yes  No  A drug can be prescribed without the involvement of a doctor.
Yes  No  A given pharmacy can be selling a given drug at several different prices.
Yes  No  Every person must have a primary doctor.
Yes  No  One doctor can be the primary doctor for several patients.
Yes  No  A doctor can also be a patient.
Yes  No  A doctor can only prescribe drugs related to their specialty.
Yes  No  A doctor can prescribe the same drug to several patients.
Yes  No  A doctor cannot be their own patient.
Question 3: Relational Languages and Queries (50 points)

Consider the following relational schema for storing information about university students and their departments. The primary key for Student is sid, the primary key for Dept is dno, and there are two foreign keys in Student: major, which references Dept, and mentor, which references Student. (Students at this university can each have another student assigned to help mentor them during their studies.)

Relations: Student(sid, sname, age, year, major, mentor)  Dept(dno, dname, chair, school)

Given this schema, write each of the following queries in the indicated relational language. Pay careful attention to the language being asked for, as no partial credit will be given if you answer a question in the wrong language. As a quick syntax reminder, to print out the names and chairs of the departments in ICS using the relational algebra or the relational calculus, you might write the following. If you are asked what a given query will print, use the example data below to compute your answer.

\[
\pi \text{ dname} \rightarrow \text{deptname}, \text{chair} \rightarrow \text{chairname} \quad (\sigma \text{ school} = 'ICS' \quad (\text{Dept}))
\]

\[
\{ t(\text{deptname}, \text{chairname}) \mid \exists d \in \text{Dept} \quad (t.\text{deptname} = d.\text{dname} \land t.\text{chairname} = d.\text{chair} \land d.\text{school} = 'ICS') \}\}

<table>
<thead>
<tr>
<th>sid</th>
<th>sname</th>
<th>age</th>
<th>year</th>
<th>major</th>
<th>mentor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sally</td>
<td>22</td>
<td>Junior</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>John</td>
<td>18</td>
<td>Freshman</td>
<td>null</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Rahul</td>
<td>22</td>
<td>Grad</td>
<td>20</td>
<td>null</td>
</tr>
<tr>
<td>4</td>
<td>Jake</td>
<td>17</td>
<td>Senior</td>
<td>30</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dno</th>
<th>dname</th>
<th>school</th>
<th>chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CS</td>
<td>ICS</td>
<td>Turing</td>
</tr>
<tr>
<td>20</td>
<td>EE</td>
<td>Eng</td>
<td>Ohm</td>
</tr>
<tr>
<td>30</td>
<td>Physics</td>
<td>PhySci</td>
<td>Einstein</td>
</tr>
</tbody>
</table>

(a) (7 pts) Print the names and ages of students who are either in their Junior year or who are majoring in a department whose chair’s name is Ohm. Write this query in the relational calculus.

\[
\exists t(\text{name}, \text{age}) \mid \exists s \in \text{Student} \quad (t.\text{name} = s.\text{sname} \land t.\text{age} = s.\text{age} \land (s.\text{year} = 'Junior' \lor \exists d \in \text{Dept} \quad (d.\text{chair} = 'Ohm' \land d.\text{dno} = s.\text{major})))
\]

SCORE: 7
**Relations:**  Student(sid, sname, age, year, major, mentor)  Dept(dno, dname, chair, school)

(b) (7 pts) Print the names, ages, and mentor names of those students who are over 20 years old, are majoring in CS, and have a mentor. Write this query in SQL.

```
SELECT s.name, s.age, m.sname
FROM Student s, Student M, Dept D
WHERE s.age > 20 AND D.dname = 'CS'
     AND D.dno = S.major AND M.sid = S.mentor
```  

(c) (7 points) Print the student ids, names, ages, years, major department numbers, and mentor ids for students who are younger than average. Write this query in SQL.

```
SELECT *
FROM Student S
WHERE S.age < (SELECT AVG(age) FROM Student)
```
Relations: Student(sid, sname, age, year, major, mentor) Dept(dno, dname, chair, school)

(d) (7 pts) Print the names and ages of students who are over 20 years old who are not majoring in EE. Write this query in the relational algebra.

\[ \pi_{\text{name}, \text{age}}(\text{Student} \setminus \text{sid} : \pi_{\text{sid}}(\text{age} \geq 20 (\text{Student}))) = \pi_{\text{sid}}(\text{Student} \land \text{major} = "EE" (\text{Dept})) \]

(Note: Using sid's to ensure correct student identification in \(\setminus\).)

(e) (7 pts) Print the total number of student majors as well as the average age of those majors for each department in the engineering (Eng) school that has at least 10 students majors. The result should have four columns containing the department’s number, name, student count, and average age, and should include one result row for each qualifying department. Write this query in SQL.

```
SELECT D.dno, D.dname, COUNT(*) AS nummajors, AVG(S.age) AS avgage
FROM Student S, Dept D
WHERE S.major = D.dno AND D.school = 'Eng'
group by D.dno, D.dname
having COUNT(*) >= 10
```
Relations: Student(sid, sname, age, year, major, mentor)  Dept(dno, dname, chair, school)

(f) (3 pts) What does the following SQL query do? Answer by showing its output for the sample data given at the start of this problem back on p. 4.

```
SELECT s.name
FROM Student s
WHERE s.major <= 10 OR s.major > 10
```

(g) (6 pts) What does the following SQL query do? Answer by showing its output for the sample data given at the start of this problem back on p. 4.

```
SELECT s.sid, s.sname, s.year, d.dname AS major, d.school
FROM (Student s LEFT OUTER JOIN Dept d ON s.major = d.dno)
WHERE NOT (s.sname LIKE ‘%e’)
```

(h) (6 pts) Which of the following statements are actual advantages of using a declarative query language? Answer by putting an X in the circle next to each such advantage.

- [ ] It leaves the selection of an appropriate data access plan up to the system.
- [ ] It allows developers to avoid having to learn SQL.
- [ ] It allows the DBA to change the physical schema without affecting database applications.
- [ ] It makes it possible to express complex search requests in a small amount of “code”.
- [x] It sounds impressive at fraternity and sorority parties.